Face Recognition Using Haar Cascade Classifier

Varun Garg
Electronics and Electrical Department
Maharaja Agrasen Institute of Technology

Kritika Garg
Information Technolgy Department
Ch. Brahm Prakash Government Engineering College

***Abstract*—with development of machine learning technology many applications have been revolutionized which earlier used to utilize high amount of resources .Face recognition is a crucial security application .Though this paper we present this application using optimized amount of resources and high efficiency.**

***Keywords*—Face detection, Machine Learning, Open CV, Raspberry Pi, Haar Cascade Classifier**

I. Introduction

The face is crucial for human identity. It is the feature which best distinguishes a person. Face recognition is an interesting and challenging problem, and impacts important applications in many areas such as identification for law enforcement, authentication for banking and security system access, and personal identification among others. Face recognition is an easy task for humans but its entirely different task for a computer. A very little is known about human recognition to date on How do we analyze an image and how does the brain encode it and Are inner features (eyes, nose, mouth) or outer features (head shape, hairline) used for a successful face recognition? Neurophysiologist David Hubel and Torsten Wiesel has shown that our brain has specialized nerve cells responding to specific local features of a scene, such as lines, edges, angles or movement. Since we don't see the world as scattered pieces, our visual cortex must somehow combine the diff erent sources of information into useful patterns.

Automatic face recognition is all about extracting those meaningful features from an image,putting them into a useful representation and performing some classifications on them.Face recognition based on the geometric features of a face is probably the most intuitive approach to Human identification.The whole process can be divided in three major steps where the first step is to find a good database of faces with multiple images for each individual.The next step is to detect faces in the database images and use them to train the face recognizer and the last step is to test the face recognizer to recognize faces it was trained for.

II. Implementation

A. Import the required modules

The Modules required to perform the facial recognition are cv2, os, image module and numpy. cv2 is the OpenCV module and contains the functions for face detection and recognition. OS will be used to maneuver with image and directory names. First, we use this module to extract the image names in the database directory and then from these names individual number is extracted, which is used as a label for the face in that image. Since, the dataset images are in gif format and as of now, OpenCV does not support gif format, Image module from PIL is used to read the image in grayscale format. Numpy arrays are used to store the images

B. Load the face detection Cascade

To Load the face detection Cascade the first step is to detect the face in each image. Once we get the region of interest containing the face in the image, we use it for training the recognizer. For the purpose of face detection, we will use the Haar Cascade provided by OpenCV. The haar cascades that come with OpenCV are located in the directory of OpenCV installation. haarcascade frontalface default.xml is used for detecting the face. Cascade is loaded using the cv2 CascadeClassifier function which takes the path to the cascade xml file. if the xml file is in the current working directory, then relative path is used.

C. Create the Face Recognizer Object

The next step involve creating the face recognizer object. The face recognizer object has functions like FaceRecognizer.train to train the recognizer and FaceRecognizer.predict to recognize a face. OpenCV currently provides Eigenface Recognizer,Fisherface Recognizer and Local Binary Patterns Histograms Face Recognizer.we have used Local Binary Patterns Histograms Face Recognizer to perform face recognition. With Local Binary Patterns it is possible to describe the texture and shape of a digital image. This is done by dividing an image into several small regions from which the features are extracted that can be used to get a measure for the similarity between the images.

D. Prepare the training set Perform the training

To create the function to prepare the training set, we will define a function that takes the absolute path to the image database as input argument and returns tuple of 2 list, one containing the detected faces and the other containing the corresponding label for that face. For example, if the ith index in the list of faces represents the 4th individual in the database, then the corresponding ith location in the list of labels has value equal to 4.

Now to perform the training using the Face Recognizer. Train function. It requires 2 arguments, the features which in this case are the images of faces and the corresponding labels assigned to these faces which in this case are the individual number that we extracted from the image names.

E. Testing

For testing the Face Recognizer, we check if the recognition was correct by comparing the predicted label predicted with the actual label actual. The label actual is extracted using the os module and the string operations from the name of the image. We also display the confidence score for each recognition.

III. DEVLOPMENT OF SECURITY SYSTEM USING FACE RECOGNITION

This security system is build to implement on a door. Where every person entered is checked and their images are taken and compared with available data. If the person is found in database, will be authorized to access otherwise access will be denied. Every image taken and their authorization whether authorized or not, will send to Google Drive with time and date of entry. From Google drive data can be getting to use to display on apps or on websites.

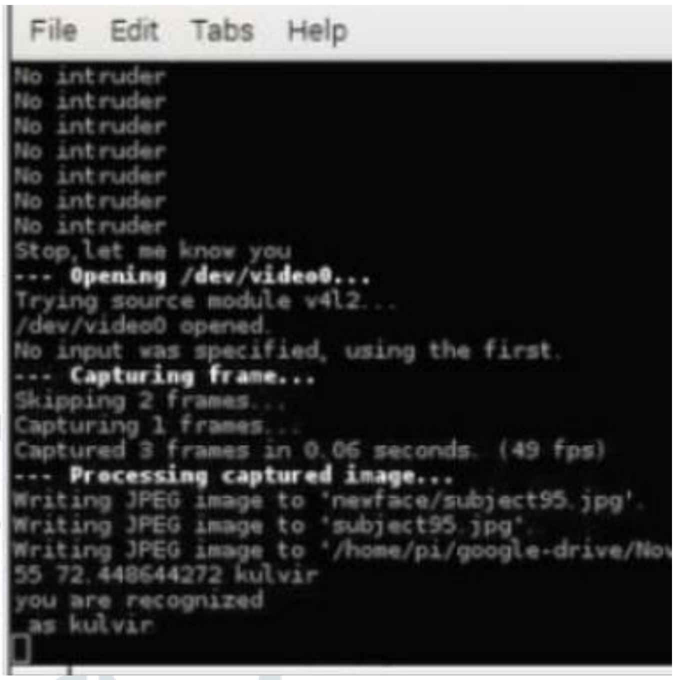
To build this security system we will require a Raspberry pi board, Camera,Sensors pair,Infrared Sensors pair,6x2 LCD screen,USB Cables, Ethernet jumper wires and Open CV module. The Raspberry Pi is a series of credit card–sized single-board computers. It could work on Linux. It has 40 GPIO pins, Ethernet port, 4 USB ports, micro SD card port etc.Camera takes the image and saves it in 'Newface' folder, from where it will be taken for comparison.A set of IR (infrared) sensors has been set, which detects the coming person.IR sensors pair sends signal to raspberry pi, which activates the camera and takes the image of person standing in front of it .LCD screen is used to display the messages. Ethernet wire is used make raspberry pi internet enable to send data to Google drive USB cable powers the raspberry pi.

A. Process of Devlopment of Security System

This is explained in steps from detecting a person to save data on Google drive. Here is given screen shots of working project and LCD displayed messages.

Loading the database: Some faces of known persons have been saved and loaded before starting the comparison.
Detecting a person: when a person is passed nearby ir sensor, raspberry pi sends signal to camera and display message on lcd to stop. Camera captures the image of person standing in front of it . "Stop, let me know you"
Comparing with saved data: After capturing it compares the image with previously saveddata. And display the message to wait. "finding.....you"

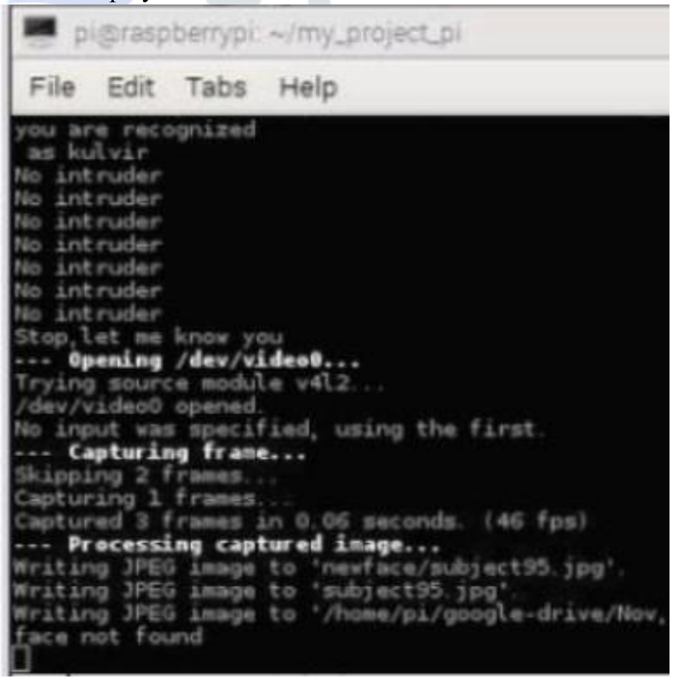
Result: If the image of coming person is present in the saved data it will print his/her name as saved. "you are recognized as ......"



```
No intruder
No intruder
No intruder
No intruder
No intruder
No intruder
No intruder
Stop,let me know you
--- Opening /dev/video0...
Trying source module v4l2...
/dev/video0 opened.
No input was specified, using the first.
--- Capturing frame...
Skipping 2 frames...
Capturing 1 frames...
Captured 3 frames in 0.06 seconds. (49 fps)
--- Processing captured image...
Writing JPEG image to 'newface/subject95.jpg'.
Writing JPEG image to 'subject95.jpg'.
Writing JPEG image to '/home/pi/google-drive/Nov
55 72.448644272 kulvir
you are recognized
 as kulvir
```

Figure 1: Screenshot of result

No intruder: when no person is detected, message displayed is "No intruder"
Face not found: If face is not detectable on image, message will be displayed "face not found"



```
you are recognized
 as kulvir
No intruder
No intruder
No intruder
No intruder
No intruder
No intruder
No intruder
Stop,let me know you
--- Opening /dev/video0...
Trying source module v4l2...
/dev/video0 opened.
No input was specified, using the first.
--- Capturing frame...
Skipping 2 frames...
Capturing 1 frames...
Captured 3 frames in 0.06 seconds. (46 fps)
--- Processing captured image...
Writing JPEG image to 'newface/subject95.jpg'.
Writing JPEG image to 'subject95.jpg'.
Writing JPEG image to '/home/pi/google-drive/Nov,
face not found
```

Figure 2: Screenshot of result

Data saved in Google drive view: Data of result outcomes of face recognition is stored on google drive, and can be get for further use on apps or website

IV. ADVANTAGES

This is real time project. It takes less than 5 seconds to give result. It is based on Linux, can be work on computer or any board with Linux in it. It could be implemented anywhere. With raspberry pi it consists of very small hardware, so easy to implement. it is affordable and the cost is very resonabe Google drive has large space. We are able to get the data on android app from Google drive. Data is available for future use. Counting of entries is done without any human help.

V. DRAWBACKS

It needs more than one image of same face, for accurate result. It needs working internet to send the data to cloud. It compares every coming image and gives positive result in confidence level, even for face is not present in data base. We need o set a threshold value below which result is acceptable. It takes time to load the database in beginning. More the database more it takes time. Raspberry pi's CPU is little bit slower than our computer's CPU.

VI. CONCLUSION

Thus using open cv librarie's haarcascade classifier we were successfully able to perform face detection with high efficency .I have used the open cv package to extract the features of face and then to compare them. Saved database should have more than one images of same face it make face recognition more accurate. Images and their corresponding labels being store in arrays. It compares every coming image with stored data and gives result in confidences level value. Even if face is not present in data it gives a confidence value. But the confidence value will be very high, so we consider a threshold value below which result is acceptable.

REFERENCES

[1] "Face Recognition using Python and OpenCV" http://hanzratech.in/2015/02/03/face-recognition-using-opencv.html.
[2] "Raspberry pi fswebcam ," http://raspberrypi.stackexchange.com/questions/8303/fswebcam-not- displaying-images-properly.
[3] "Internet of things," http://micrium.com/designingtheinternetofthingspart1iot-devicesandlocalnetworks/
[4] "Installing Google Drive On Raspberry Pi," http://www.home-automation-community.com/google-drive-on-raspberry-pi/".
[5] "open cv on raspberry pi B+," http://www.pyimagesearch.com/2015/06/22/install-opencv-3-0- and-python-2-7-on-ubuntu/ .
[6] "Python Software foundation(us)," https://www.python.org/.
[7] Raspberry pi, documentation, http://venturebeat.com/2014/09/02/were-going-to-need-more- storage-for-the-internet-of-things/.
[8] "Storage need in iot " http://venturebeat.com/2014/09/02/were-going-to-need-more-storage-for- the-internet-of-things/